Low Level Design

PCA (Customer Personality Analysis)

|  |  |
| --- | --- |
| Written By | Shashank Shukla |
| Document Version | 0.1 |
| Last Revised Date | 07-07-2023 |

**Document Control**

**Change Record:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.1 | 07-07-2023 | Shashank Shukla | Introduction and architecture define |
|  |  |  |  |
|  |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
| 0.2 | 21 – May -  2021 | Khusali | Document Content, Version Control and Unit Test Cases to be added |

**Approval Status:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review**  **Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

Contents

1. **Introduction** 1
   1. **What is Low-Level design document?** 1
   2. **Scope** 1
2. **Architecture** 2
3. **Architecture Description** 3
   1. **Data Description** 3
   2. **Web Scrapping** 3
   3. **Data Transformation** 3
   4. **Data Insertion into Database** 3
   5. **Export Data from Database** 3
   6. **Data Pre-processing** 3
   7. **Data Clustering** 3
4. **Model Building** 4
5. **Data from User** 4
6. **Data Validation** 4
7. **User Data Inserting into Database** 4
8. **Data Clustering** 4
9. **Model Call for Specific Cluster** 4
10. **Recipe Recommendation & Saving Output in Database** 4
11. **Deployment** 4
12. **Unit Test Cases** 5

# **Introduction**

* 1. **What is Low-Level design document?**

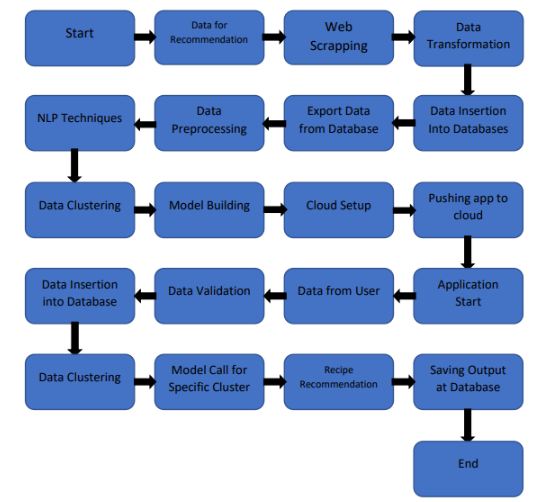
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

* 1. **Scope**

Low-level design (LLD) is a component-level design process that follows a step-by-

step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

1. **Architecture**



1. Architecture Description
   1. Data Description
   2. Data Transformation
   3. Data Insertion into Database
   4. Export Data from Database
   5. Data Pre-processing
   6. Data Clustering
   7. Model Building
   8. Deployment
2. Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads completely for the user when the URL is accessed | 1. Application URL is accessible 2. Application is deployed | The Application should load completely for the user when the URL is accessed |
| Verify whether the User is able to sign  up in the application | 1. Application is  accessible | The User should be able to sign up  in the application |
| Verify whether user is able to successfully login to the application | 1. Application is accessible 2. User is signed up to the application | User should be able to successfully login to the application |
| Verify whether user is able to see input fields on logging in | 1. Application is accessible 2. User is signed up to the application 3. User is logged in   to the application | User should be able to see input fields on logging in |
| Verify whether user is able to edit all input fields | 1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application | User should be able to edit all input fields |
| Verify whether user gets Submit button to submit the inputs | 1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application | User should get Submit button to submit the inputs |
| Verify whether user is presented with recommended results on clicking  submit | 1. Application is accessible 2. User is signed up to the application 3. User is logged in   to the application | User should be presented with recommended results on clicking  submit |
| Verify whether the recommended results are in accordance to the selections user made | 1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application | The recommended results should be in accordance to the selections user made |
| Verify whether user has options to filter the recommended results as well | 1. Application is accessible 2. User is signed up | User should have options to filter the recommended results as well |